breath and spirit of the present invention as disclosed herein.

CLAIMS

What is claimed is:

1. A method for controlling the transmission of data packets through a network by controlling a Transmission Control Protocol (TCP) rate in a network device having a shared buffer with shared buffer space, the method comprising:

organizing a forward data buffer into one or more queues that store at least one forward data packet;

calculating the network device's advertised window size by implementing an integral control algorithm that uses information pertaining to the one or more queues;

providing the network device's advertised window size to a TCP source; and

calculating a dynamic buffer threshold based, at least in part, upon the sum of the queue sizes and the shared buffer space.

2. The method of claim 1 wherein the step of organizing a forward data buffer further comprises:

organizing the forward data buffer into one or more queues with one queue per service class.

3. The method of claim 1 wherein the at least one forward data packet is stored according to its service class.

4. The method of claim 1 wherein the step of calculating a network device's advertised window size further comprises:

initializing a timer to a predetermined time interval Δt, and an iteration counter to a predetermined initial value n;

sampling a current queue size $q_i(n)$ during the predetermined time interval Δt;

calculating a current error signal $e_i(n)$ based, at least in part, upon the current queue size $q_i(n)$;

calculating the network device's advertised window size $W_i(n)$, based, at least in part, upon the current error signal $e_i(n)$ according to the equation: $W_i(n) = \left[W_i(n-1) + \alpha e_i(n)\right]_{W_{\min}}^{W_{\max}}$, where α, $W_{\max}$, and $W_{\min}$, are predetermined parameters;

resetting the timer, upon expiration of the predetermined interval Δt; and

iterating the iteration counter, upon expiration of the predetermined time interval Δt.

5. The method of claim 4 wherein the steps of calculating a current error signal $e_i(n)$ and calculating

the network device's advertised window size further comprise:

filtering the current error signal $e_i(n)$ according to the relation: $\hat{e}_i(n) = (1-\beta)\hat{e}_i(n-1) + \beta e_i(n)$, where β is a predetermined parameter; and

calculating the network device's advertised window size $W_i(n)$, based, at least in part, upon the filtered current error signal $\hat{e}_i(n)$ according to the equation: $W_i(n) = \left[W_i(n-1) + \alpha\hat{e}_i(n)\right]_{W_{min}}^{W_{max}}$, where α, W_{max}, and W_{min}, are predetermined parameters.

6. The method of claim 1 wherein the step of providing the network device's advertised window size to a TCP source further comprises:

carrying information relating to the network device's advertised window size by returning TCP acknowledgements in a receiver's advertised window field.

7. The method of claim 1 wherein the step of providing the network device's advertised window size to a TCP source further comprises:

updating a TCP receiver's advertised window size.

8. The method of claim 7 wherein the step of updating a TCP receiver's advertised window size further comprises:

identifying whether a packet is an ACK packet, and, if not, putting the non-ACK packet in a reverse data buffer;

determining a service class for the identified ACK packet;

reading the TCP receiver's advertised window size (RW_{rec}) and a checksum ($RCHKSUM$) from the identified ACK packet;

determining whether the TCP receiver's advertised window size RW_{REC}, is less than or equal to the calculated network device's advertised window size $W_i(n)$ and, if not setting a advertised window field in the identified ACK packet equal to the network device's advertised window size $W_i(n)$ and updating the checksum field for the identified ACK packet.

9. The method of claim 1 wherein the step of calculating a dynamic buffer threshold further comprises:

initializing a timer to a predetermined time interval Δs and an iteration counter to a predetermined initial value n;

setting an initial dynamic buffer threshold $T(0)$ equal to a parameter γ multiplied by a buffer size B and divided by a number of service classes K;

sampling a current queue size $q_i(n)$ during the predetermined time interval Δs;

calculating a sum of the sampled current queue size according to the equation: $Q(n) = \sum_{i=1}^{K} q_i(n)$;

determining whether the sum of the sampled current queue size is less than the product of the parameter and the buffer size γB;

if so, updating the dynamic buffer threshold

according to $\min\{T(n-1)+\Delta T,\gamma B\}$, where ΔT is a step size that controls the rate at which the dynamic buffer threshold changes;

if not, updating the dynamic buffer threshold according to $\max\{T(n-1)-\Delta T,T_{\min}\}$, where $T_{\min}$ is a predetermined minimum size for the dynamic buffer threshold;

resetting the timer, upon expiration of the predetermined interval Δs; and

iterating the iteration counter, upon expiration of the predetermined time interval Δs.

10. The method of claim 9 wherein the step of calculating a sum of the sampled current queue size further comprises:

filtering the sum of the sampled current queue size $Q(n)$ according to the relation: $\hat{Q}(n)=(1-\varphi)\hat{Q}(n-1)+\varphi Q(n)$, wherein φ is a predetermined parameter.

11. An apparatus for controlling the transmission of data packets through a network by controlling a Transmission Control Protocol (TCP) rate in a network device having a shared buffer with shared buffer space, the apparatus comprising:

a forward data buffer, organized into one or more queues that store at least one forward data packet;

a network device's advertised window size calculation module that calculates a network

device's advertised window size by implementing an integral control algorithm that uses information pertaining to the one or more queues;

a feed back module that provides the network device's advertised window size to a TCP source; and

a dynamic buffer threshold module that calculates a dynamic buffer threshold based, at least in part, upon the sum of the queue sizes and the shared buffer space.

12. The apparatus of claim 11 wherein the network device's advertised window size calculation module further comprises:

a timer, initially set to a predetermined time interval Δt, and an iteration counter initially set to a predetermined initial value n;

a current queue size sampler that samples a current queue size $q_i(n)$ during the predetermined time interval Δt;

a current error signal calculation module that calculates a current error signal $e_i(n)$ based, at least in part, upon the current queue size $q_i(n)$;

a window size calculation module that calculates the network device's advertised window size $W_i(n)$, based, at least in part, upon the current error signal $e_i(n)$ according to the equation:

$W_i(n) = \left[W_i(n-1) + \alpha e_i(n)\right]_{W_{min}}^{W_{max}}$, where α, W_{max}, and W_{min},

are predetermined parameters.

13. The apparatus of claim 12 wherein the current error signal calculation module further comprises:

a filter module that filters the current error signal $e_i(n)$ according to the relation: $\hat{e}_i(n) = (1-\beta)\hat{e}_i(n-1) + \beta e_i(n)$, where β is a predetermined parameter; and

wherein the window size calculation module calculates the network device's advertised window size $W_i(n)$, based, at least in part, upon the filtered current error signal $\hat{e}_i(n)$ according to the equation: $W_i(n) = \left[W_i(n-1) + \alpha \hat{e}_i(n)\right]_{W_{\min}}^{W_{\max}}$, where α, $W_{\max}$, and $W_{\min}$, are predetermined parameters.

14. The apparatus of claim 11 wherein the feed back module further comprises:

an advertised window size updating module that updates a TCP receiver's advertised window size.

15. The apparatus of claim 14 wherein the advertised window size updating module further comprises:

an ACK packet identification module that identifies whether a packet is an ACK packet, and, if not, puts the non-ACK packet in a reverse data buffer;

an ACK packet classifier that determines a service class for the identified ACK packet;

an advertised window size reader that reads a TCP

receiver's advertised window size (RW_{rec}) and a checksum ($RCHKSUM$) from the identified ACK packet;

a window size comparison module that determines whether the TCP receiver's advertised window size RW_{REC}, is less than or equal to the calculated network device's advertised window size $W_i(n)$ and, if not sets an advertised window field in the identified ACK packet equal to the calculated network device's advertised window size $W_i(n)$ and updates the checksum field for the identified ACK packet.

16. The apparatus of claim 11 wherein the dynamic buffer threshold module further comprises:

a timer initially set to a predetermined time interval Δs and an iteration counter initially set to a predetermined initial value n;

a current queue size sampler that samples a current queue size $q_i(n)$ during the predetermined time interval Δs;

a current queue size calculation module that calculates a sum of the sampled current queue size according to the equation: $Q(n) = \sum_{i=1}^{K} q_i(n)$, where K is a number of service classes;

a dynamic buffer threshold determiner that determines whether the sum of the sampled current queue size is less than the product of a parameter γ and a buffer size B;

and an updating module that updates the dynamic

buffer threshold if the sum of the sampled current queue size is less than the product of the parameter γ and the buffer size B, according to $\min\{T(n-1)+\Delta T,\gamma B\}$, where ΔT is a step size that controls the rate at which the dynamic buffer threshold changes and if the sum of the sampled current queue size is not less than the product of the parameter γ and a buffer size B, updates the dynamic buffer threshold according to $\max\{T(n-1)-\Delta T,T_{\min}\}$, where $T_{\min}$ is a predetermined minimum size for the dynamic buffer threshold.

17. The apparatus of claim 16 wherein the current queue size calculation module further comprises:

a filter that filters the sum of the sampled current queue size $Q(n)$ according to the relation: $\hat{Q}(n)=(1-\varphi)\hat{Q}(n-1)+\varphi Q(n)$, wherein φ is a predetermined parameter.

18. An article of manufacture for controlling the transmission of data packets through a network by controlling a Transmission Control Protocol (TCP) rate in a network device having a shared buffer with shared buffer space, the article of manufacture comprising:

at least one processor readable carrier; and

instructions carried on the at least one carrier;

wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

organize a forward data buffer into one or more queues that store at least one forward data packet;

calculate a network device's advertised window size by implementing an integral control algorithm that uses information pertaining to the one or more queues;

provide the network device's advertised window size to a TCP source; and

calculate a dynamic buffer threshold based, at least in part, upon the sum of the queue sizes and the shared buffer space.

19. The article of manufacture of claim 18 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

organize the forward data buffer into one or more queues with one queue per service class.

20. The article of manufacture of claim 18 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

store the at least one forward data packet according to its service class.

21. The article of manufacture of claim 18 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

initialize a timer to a predetermined time interval

Δt, and an iteration counter to a predetermined initial value n;

sample a current queue size $q_i(n)$ during the predetermined time interval Δt;

calculate a current error signal $e_i(n)$ based, at least in part, upon the current queue size $q_i(n)$;

calculate the network device's advertised window size $W_i(n)$, based, at least in part, upon the current error signal $e_i(n)$ according to the equation: $W_i(n)=\left[W_i(n-1)+\alpha e_i(n)\right]_{W_{\min}}^{W_{\max}}$, where α, $W_{\max}$, and $W_{\min}$, are predetermined parameters;

reset the timer, upon expiration of the predetermined interval Δt; and

iterate the iteration counter, upon expiration of the predetermined time interval Δt.

22. The article of manufacture of claim 21 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

filter the current error signal $e_i(n)$ according to the relation: $\hat{e}_i(n)=(1-\beta)\hat{e}_i(n-1)+\beta e_i(n)$, where β is a predetermined parameter; and

calculate the network device's advertised window size $W_i(n)$, based, at least in part, upon the filtered current error signal $\hat{e}_i(n)$ according to

the equation: $W_i(n) = \left[W_i(n-1) + \alpha \hat{e}_i(n)\right]_{W_{\min}}^{W_{\max}}$, where α, $W_{\max}$, and $W_{\min}$, are predetermined parameters.

23. The article of manufacture of claim 18 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

carry information relating to the network device's advertised window size by returning TCP acknowledgements in a receiver's advertised window field.

24. The article of manufacture of claim 18 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

update a TCP receiver's advertised window size.

25. The article of manufacture of claim 24 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

identify whether a packet is an ACK packet, and, if not, put the non-ACK packet in a reverse data buffer;

determine a service class for the identified ACK packet;

read a TCP receiver's advertised window size (RW_{rec}) and a checksum ($RCHKSUM$) from the identified ACK packet;

determine whether the TCP receiver's advertised

window size RW_{REC}, is less than or equal to the calculated network device's advertised window size $W_i(n)$ and, if not setting an advertised window field in the identified ACK packet equal to the calculated network device's advertised window size $W_i(n)$ and updating the checksum field for the identified ACK packet.

26. The article of manufacture of claim 18 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

initialize a timer to a predetermined time interval Δs and an iteration counter to a predetermined initial value n;

set an initial dynamic buffer threshold $T(0)$ equal to a parameter γ multiplied by a buffer size B and divided by a number of service classes K;

sample a current queue size $q_i(n)$ during the predetermined time interval Δs;

calculate a sum of the sampled current queue size according to the equation: $Q(n)=\sum_{i=1}^{K} q_i(n)$;

determine whether the sum of the sampled current queue size is less than the product of the parameter and the buffer size γB;

if so, updating the dynamic buffer threshold according to $\min\{T(n-1)+\Delta T, \gamma B\}$, where ΔT is a step size that controls the rate at which the dynamic buffer threshold changes;

if not, updating the dynamic buffer threshold according to $\max\{T(n-1)-\Delta T, T_{\min}\}$, where $T_{\min}$ is a predetermined minimum size for the dynamic buffer threshold;

reset the timer, upon expiration of the predetermined interval Δs; and

iterate the iteration counter, upon expiration of the predetermined time interval Δs.

27. The article of manufacture of claim 26 wherein the instructions are configured to be readable from the at least one carrier by at least one processor and thereby cause the at least one processor to operate so as to:

filter the sum of the sampled current queue size $Q(n)$ according to the relation: $\hat{Q}(n) = (1-\varphi)\hat{Q}(n-1) + \varphi Q(n)$, wherein φ is a predetermined parameter.

28. A signal embodied in a carrier wave and representing sequences of instructions which, when executed by at least one processor, cause the at least one processor to control the transmission of data packets through a network by controlling a Transmission Control Protocol (TCP) rate in a network device having a shared buffer with shared buffer space, by performing the steps of:

organizing a forward data buffer into one or more queues that store at least one forward data packet;

calculating a network device's advertised window size by implementing an integral control algorithm that uses information pertaining to

the one or more queues;

providing the network device's advertised window size to a TCP source; and

calculating a dynamic buffer threshold based, at least in part, upon the sum of queue sizes and the shared buffer space.

29. The signal of claim 28 wherein the step of organizing a forward data buffer further comprises:

organizing the forward data buffer into one or more queues with one queue per service class.

30. The signal of claim 28 wherein the at least one forward data packet is stored according to its service class.

31. The signal of claim 28 wherein the step of calculating a network device's advertised window size further comprises:

initializing a timer to a predetermined time interval Δt, and an iteration counter to a predetermined initial value n;

sampling a current queue size $q_i(n)$ during the predetermined time interval Δt;

calculating a current error signal $e_i(n)$ based, at least in part, upon the current queue size $q_i(n)$;

calculating the network device's advertised window size $W_i(n)$, based, at least in part, upon the current error signal $e_i(n)$ according to the

equation: $W_i(n) = \left[W_i(n-1) + \alpha e_i(n)\right]_{W_{min}}^{W_{max}}$, where α, W_{max}, and W_{min}, are predetermined parameters;

resetting the timer, upon expiration of the predetermined interval Δt; and

iterating the iteration counter, upon expiration of the predetermined time interval Δt.

32. The signal of claim 31 wherein the steps of calculating a filtered current error signal $e_i(n)$ and calculating the network device's advertised window size further comprise:

filtering the current error signal $e_i(n)$ according to the relation: $\hat{e}_i(n) = (1-\beta)\hat{e}_i(n-1) + \beta e_i(n)$, where β is a predetermined parameter; and

calculating the network device's advertised window size $W_i(n)$, based, at least in part, upon the filtered current error signal $\hat{e}_i(n)$ according to the equation: $W_i(n) = \left[W_i(n-1) + \alpha \hat{e}_i(n)\right]_{W_{min}}^{W_{max}}$, where α, W_{max}, and W_{min}, are predetermined parameters.

33. The signal of claim 28 wherein the step of providing the network device's advertised window size to a TCP source further comprises:

carrying information relating to the network device's advertised window size by returning TCP acknowledgements in a receiver's advertised window field.

34. The signal of claim 28 wherein the step of providing the network device's advertised window size to a TCP

source further comprises:

updating a TCP receiver's advertised window size.

35. The signal of claim 34 wherein the step of updating a TCP receiver's advertised window size further comprises:

identifying whether a packet is an ACK packet, and, if not, putting the non-ACK packet in a reverse data buffer;

determining a service class for the identified ACK packet;

reading a TCP receiver's advertised window size (RW_{rec}) and a checksum ($RCHKSUM$) from the identified ACK packet;

determining whether the TCP receiver's advertised window size RW_{REC}, is less than or equal to the calculated network device's advertised window size $W_i(n)$ and, if not setting an advertised window field in the identified ACK packet equal to the calculated network device's advertised window size $W_i(n)$ and updating the checksum field for the identified ACK packet.

36. The signal of claim 28 wherein the step of calculating a dynamic buffer threshold further comprises:

initializing a timer to a predetermined time interval Δs and an iteration counter to a predetermined initial value n;

setting an initial dynamic buffer threshold $T(0)$ equal to a gain constant γ multiplied by a buffer size B and divided by a number of

service classes K;

sampling a current queue size $q_i(n)$ during the predetermined time interval Δs;

calculating a sum of the sampled current queue size according to the equation: $Q(n)=\sum_{i=1}^{K} q_i(n)$;

determining whether the sum of the sampled current queue size is less than the product of the gain constant and the buffer size γB;

if so, updating the dynamic buffer threshold according to $\min\{T(n-1)+\Delta T, \gamma B\}$, where ΔT is a step size that controls the rate at which the dynamic buffer threshold changes;

if not, updating the dynamic buffer threshold according to $\max\{T(n-1)-\Delta T, T_{\min}\}$, where $T_{\min}$ is a predetermined minimum size for the dynamic buffer threshold;

resetting the timer, upon expiration of the predetermined interval Δs; and

iterating the iteration counter, upon expiration of the predetermined time interval Δs.

37. The signal of claim 36 wherein the step of calculating a sum of the sampled current queue size further comprises:

filtering the sum of the sampled current queue size $Q(n)$ according to the relation: $\hat{Q}(n)=(1-\varphi)\hat{Q}(n-1)+\varphi Q(n)$, wherein φ is a predetermined parameter.

38. A computer data signal embodied in a carrier wave readable by a computing system and encoding a computer program of instructions for executing a computer process performing the method recited in claim 1.